VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

NAHRÁVÁNÍ TELEFONÁTU A VYHLEDÁVÁNÍ PRO SKYPE

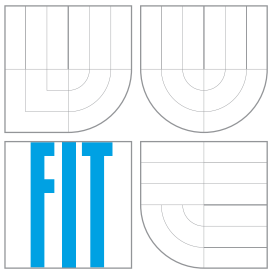
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JIŘÍ NYTRA

BRNO 2011

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

NAHRÁVÁNÍ TELEFONÁTU A VYHLEDÁVÁNÍ PRO SKYPE

VOICE RECORDING AND SEARCH FOR SKYPE

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

JIŘÍ NYTRA

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. PETR SCHWARZ, Ph.D.

BRNO 2011

Abstrakt

Práce se zabývá vytvořením programu komunikujícím s aplikací Skype, který umožňuje nahrávat hovory, v nichž dokáže vyhledávat klíčová slova pomocí moderních technologií rozpoznávání řeči. V práci je představeno rozhraní a protokol pro komunikaci s programem Skype, nahrávání hovoru a metoda LVCSR pro vyhledávání klíčových slov.

Abstract

This work deals with the creation of a program communicating with Skype, which provides record calls in which can search for keywords by using advanced speech recognition technology. The work is presented and the interface protocol to communicate with Skype, call recording and method LVCSR for searching keywords.

Klíčová slova

aplikace Skype, zaznamenávání hovorů, rozpoznávání řeči, vyhledávání, LVCSR, wxWidgets

Keywords

Skype aplication, call recording, speech recognition, searching, LVCSR, wxWidgets

Citace

Jiří Nytra: Nahrávání telefonátu a vyhledávání
pro Skype, bakalářská práce, Brno, FIT VUT v Brně, 2011

Nahrávání telefonátu a vyhledávání pro Skype

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Petr Schwarze Ph.D. Další informace mi poskytl Ing. Tomáš Cipr. Dále prohlašuji, že jsem uvedl všechny literární prameny a publikace ze kterých sem čerpal.

.....
Jiří Nytra
17. května 2011

Poděkování

Velmi děkuji vedoucímu bakalářské práce panu Ing. Petr Schwarzovi Ph.D za trpělivost a pomoc při řešení problémů. Dále bych chtěl poděkovat celé řečové skupině FIT VUT, zejména pak panu Ing. Tomáši Ciprovi za velmi cenné rady.

Obsah

Úvod

Cílem celého projektu je vytvořit modul pro komunikační program Skype, který umožní nahrávat telefonní hovory a vyhledávat v nich klíčová slova. Projekt se snaží usnadnit život lidem, kteří denně používají Skype. Každý z těchto uživatelů má občas potřebu právě probíhající hovor uložit a poslechnout si jej později. Hovor většinou obsahuje informace pro uživatele důležité, které by byl nucen zaznamenat jinak. Příkladem mohou být důležitá pracovní jednání, sjednání termínu a místa schůzky, popřípadě popis cesty.

Po určité době má uživatel mnoho nahrávek a velmi těžce se v nich orientuje. Je tedy nutné řešit tento problém a zvolit co nejpřehlednější strukturu zobrazení zaznamenaných hovorů. Projekt se rovněž zabývá filtrováním a seřazením nahrávek dle různých atributů.

Zcela jistě existuje mnoho programů, pomocí kterých je možné zaznamenávat hovory. Avšak pro rychlé vyhledání potřebných informací samotný záznam nestačí. K jejich nalezení je nutné si celou nahrávku poslechnout, což je časově velmi náročné. Proto se projekt snaží pomocí moderních technologií, vyhledáváním klíčových slov v hlase, zrychlit tuto operaci.

Tento dokument se dělí na kapitoly, přičemž každá kapitola popisuje určitou část projektu. První kapitola detailně popisuje a rozebírá problémy projektu. Zároveň je zde uveden popis a zhodnocení cizích projektů zabývajících se podobným tématem. V následující kapitole je popsáno řešení a implementace knihovny zajišťující komunikaci s programem Skype a nahrávání hovorů. V předposlední třetí kapitole popisuji realizaci uživatelské grafické aplikace využívající možnosti vytvořené knihovny. Taktéž je zde popsána funkce vyhledávání klíčových slov v záznamech hovorů. Následuje závěrečná kapitola, v níž shrnuji projekt jako celek a uvádím možnosti pro jeho další vývoj.

Kapitola 1

Analýza problémů a principy jejich řešení

Protože komunikace s programem Skype a následné nahrávání hovorů, v nichž lze vyhledávat klíčová slova, nejsou triviální problémy, podíváme se na ně v této kapitole podrobněji. Jelikož projekt řeší dva větší problémy, dělí se kapitola na dvě části. V první části je analyzován problém záznamu hovorů ze Skypu. Problém vyhledávání klíčových slov v hlase rozebírá druhá část.

1.1 Nahrávání hovorů prostřednictvím aplikace Skype

Tato část se věnuje analýze nahrávání hovorů uskutečněných prostřednictvím programu Skype. V úvodu je krátce představen vznik a historie samotné aplikace Skype. Následuje popis rozhraní, pomocí něhož lze se Skypem komunikovat. Na konci jsou představeny a zhodnoceny projekty zabývající se stejným tématem.

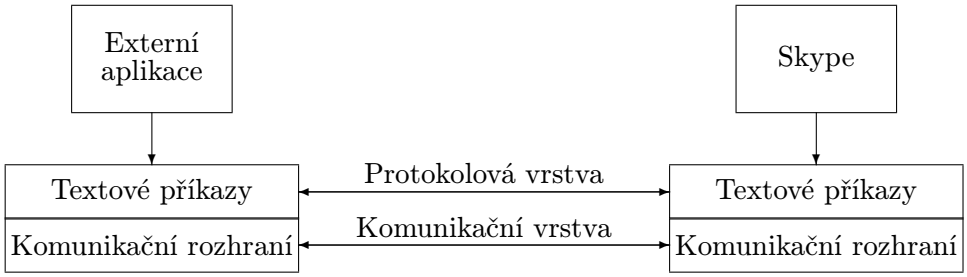
1.1.1 Aplikace Skype

Skype je komunikační program založený na peer-to-peer technologii. Umožňuje provozovat internetovou telefonii (VoIP), video hovory, zasílaní textových zpráv (Instant messaging) nebo přenos souborů. Autoři *Niklas Zennström* a *Janus Friis*, původem z Estonska, vyvinuly tuto aplikaci a založili společnost ve státě Luxembourg v roce 2003 [3]. Od této doby prošla aplikace mnoha změnami a získala si přízeň mnoha uživatelů. Svědčí o tom počet registrovaných účtů, který se v roce 2010 dostal přes hranici 650 milionů [7].

1.1.2 Aplikační rozhraní Skype

Skype API (Application Programming Interface) slouží jako prostředek pro ovládání programu Skype pomocí externích skriptů, aplikaci nebo zařízení. Tyto externí prvky mohou dále rozšiřovat možnosti programu Skype o nové funkce. Skype API se dělí na dvě základní vrstvy. Komunikační vrstvu a protokolovou vrstvu. Schéma komunikačního modelu Skype API znázorňuje obr. 1.1.

Komunikační vrstva zodpovídá za navázání, udržování a ukončení spojení mezi aplikací Skype a externím zařízením. Tato vrstva je platformě závislá. Závislost vyplývá z různých



Obrázek 1.1: Schéma komunikačního modelu aplikačního rozhraní programu Skype

technik komunikace mezi aplikacemi na platformách Windows, Linux a Mac.

Komunikace na platformě Windows probíhá pomocí systému zasílání zpráv mezi okny aplikace (Window message). K ustavení spojení mezi externí aplikaci a programem Skype se používají tyto zprávy:

- `SkypeControlAPIDiscover`
- `SkypeControlAPIAttach`

Pro navázaní komunikace vyšle klient (externí aplikace) všem okolním aplikacím zprávu `SkypeControlAPIDiscover` (žádost o ustavení spojení), kde v parametrech této zprávy zaznamená svůj identifikátor. Je-li spuštěn program Skype, ihned na tuto zprávu zareaguje. Jako svou reakci zašle klientovi pomocí identifikátoru zprávu `SkypeControlAPIAttach`. Tato zpráva obsahuje jednu z následujících odpovědí:

- Spojení navázáno v pořádku. V tomto případě zpráva obsahuje identifikátor aplikace Skype.
- Skype čeká na potvrzení komunikace od uživatele.
- Uživatel explicitně odmítl komunikovat této externí aplikaci s programem Skype.
- Spojeni není možné navázat. Tato situace může nastat například v případě, že uživatel není přihlášen do aplikace Skype.

Je-li spojení navázáno úspěšně, externí aplikace a program Skype mohou mezi sebou komunikovat. Komunikace probíhá zasíláním zpráv za pomocí identifikátorů, které si aplikace vyměnily při ustavení spojení.

V případě, že služba Skype není vůbec spuštěna, neobdrží klient na žádost o připojení žádnou odpověď. Pro zjištění dostupnosti rozhraní Skype by klient musel periodicky zasílat žádost o připojení. Důsledkem by bylo zbytečné zatížení systému. Proto vždy při spuštění rozešle aplikace Skype všem okolním programům zprávu o jeho dostupnosti. Ti se pak mohou pokusit o připojení.

Komunikace externí aplikace s programem Skype na platformě Linux lze zajistit dvěma způsoby. V prvním případě lze využít systém zasílání zpráv D-BUS (D-BUS messaging). Druhou možností je použít zasílání zpráv prostřednictvím X serveru (X11 massaging). Na

platformě MAC je komunikace zajištěna pomocí jednoho ze systémů s názvy Cocoa, Carbon nebo AppleScript. Jelikož se projekt věnuje především návrhu a realizaci pro platformu Windows, nebudu výše zmíněné systémy dále rozebírat.

Protokolová vrstva představuje prostředek pro dorozumívání mezi externí aplikací a programem Skype. Jedná se o jednoduchý textový protokol, který obsahuje soubor příkazů, pomocí nichž spolu aplikace hovoří. Všechny textové příkazy protokolu jsou zasílány ve znakové sadě UTF-8. Příkaz, který odesílá klient programu Skype, se dělí na dvě části a jeho schéma vypadá následovně:

```
#<id_příkazu> příkaz
```

Příkaz je uvozen znakem `#`. Následuje volitelný alfa-numerický identifikátor příkazu a nakonec vlastní příkaz. Na tento příkaz Skype reaguje odpovědí, v následujícím tvaru:

```
#<id_příkazu> odpověď | chyba
```

Odpověď opět obsahuje identifikátor příkazu. Za ním nasleduje odpověď na dotaz klienta nebo (v případě nesprávného dotazu) číslo chyby. Pro lepší představu o průběhu komunikace zde uvedu pár příkladů. Symbol `->` znázorňuje příkaz zaslaný klientem programu Skype. Opačný symbol `<-` zprávu zaslanou Skypem klientovi [1].

- Jednoduchý dotaz a odpověď

```
-> #AB GET USERSTATUS
<- #AB USERSTATUS ONLINE
```

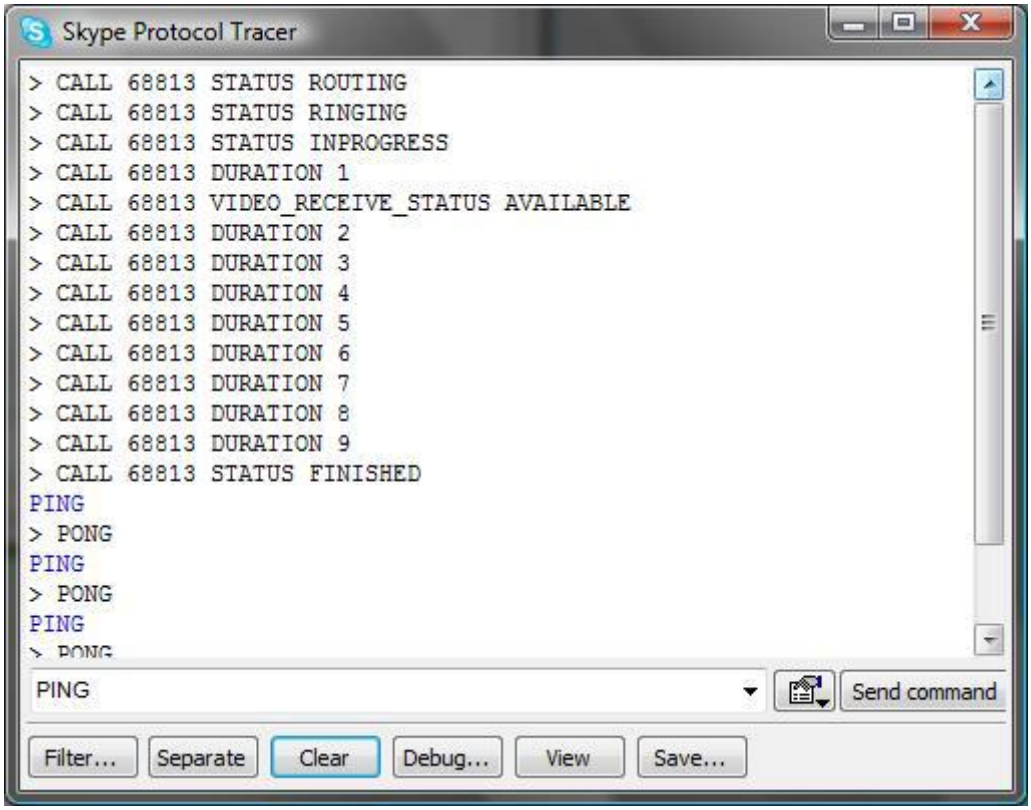
- Chybný dotaz a oznámení chyby v odpovědi

```
-> #123 GET XYZ
<- #123 ERROR 7 GET: invalid WHAT
```

- Nastavení uživatelského statusu

```
-> #abc SET USERSTATUS ONLINE
<- #abc USERSTATUS ONLINE // potvrzení přijetí příkazu abc
<- USERSTATUS ONLINE // příkaz byl zpracován
```

Pro účely vývoje je k dispozici program pro testovaní komunikaci se Skypem `tracker.exe`. Aplikace na jednotlivých řádcích zobrazuje probíhající komunikaci. V dolní části programu lze nalézt textové pole, do něhož může uživatel zadat libovolný příkaz, a ten zaslat Skypu. Viz obr. 1.2.



Obrázek 1.2: Obrázek jednoduché aplikace tracker.exe pro sledovaní komunikace s programem Skype.

1.1.3 Problém nahrávání hovorů

Nahrávání hovorů uskutečněných pomocí Skype je jedním ze základních problémů řešených v tomto projektu. Řešení problému se dělí na několik částí:

- Zjištění zahájení příchozího nebo odchozího hovoru.
- Získaní zvukových dat právě probíhajícího hovoru.
- Uložení dat v určitém formátu do zvoleného souboru a adresáře.
- Zjištění ukončení hovoru.

K řešení zmíněných částí problému nahrávání hovoru lze s výhodou použít rozhraní programu Skype rozebrané v předchozí kapitole. V následujících odstavcích budou představena možná řešení výše uvedených částí.

Ke zjištění zahájení a ukončení komunikace lze využít příkazy definované v referenčním manuálu *Skype public API* [1]. Následující příklad znázorňuje typickou posloupnost příkazů, kterou zasílá program Skype všem klientům pří zahájení a ukončení hovoru.

```
-> CALL 1403 STATUS INPROGRESS
-> CALL 1403 DURATION 1
-> CALL 1403 DURATION 2
```

```
-> CALL 1403 DURATION 3
-> CALL 1403 STATUS FINISHED
```

Za začátek hovoru můžeme považovat příkaz objektu `CALL` se statusem `INPROGRESS`. Status `FINISHED` stejného objektu představuje konec hovoru. Během hovoru vysílá Skype periodicky zprávy o uplynulé době hovoru.

Pro získání zvukových dat nahrávky lze taktéž využít možností veřejného rozhraní aplikace Skype. Jedná se opět o příkazy objektu `CALL` v následujícím tvaru:

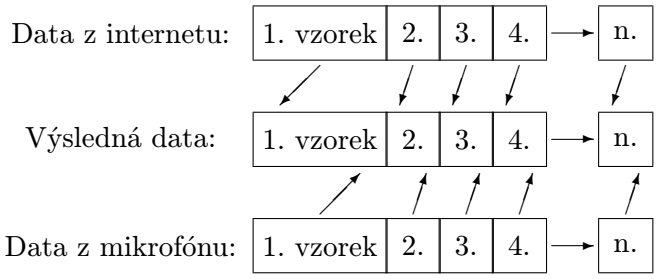
```
ALTER CALL <id> SET_OUTPUT SOUNDCARD="implicitní" |
                           PORT="číslo portu" |
                           FILE="cesta k souboru"

ALTER CALL <id> SET_CAPTURE_MIC PORT="číslo portu" |
                                FILE="cesta k souboru"
```

První příkaz slouží k získání dat přicházejících prostřednictvím internetu (data, která vyprodukovala osoba, se kterou komunikujeme). O získání dat z mikrofónu našeho PC (data, která jsme vyprodukovali my) se stará příkaz druhý. U obou příkazů je možné nastavit jeden ze dvou různých způsobů distribuce dat.

- Nastavení parametru `PORT` – program Skype zasílá data prostřednictvím TCP/IP protokolu na zvolený port. Data jsou zasílána ve formátu PCM RAW mono se vzorkovací frekvencí 16 KHz při 16 bitech na vzorek.
- Nastavení parametru `FILE` – program Skype automaticky uloží data do zvoleného souboru ve formátu WAV se stejnými parametry jako v předchozím případě.

Poslední částí problému nahrávání hovoru je samotné uložení zvukových dat do výsledného souboru. Z předchozího odstavce je zřejmé, že aplikace Skype odděluje data přicházejících z mikrofónu a data přicházející z internetu. Ve výsledku však požadujeme jeden soubor, který bude složený z obou signálů. Jednou z možností je uložit data do dvou různých kanálů, a tak vytvořit výslednou stereo nahrávku. Postup sloučení je znázorněn na obrázku 1.3.



Obrázek 1.3: Sloučení dat do výsledného formátu

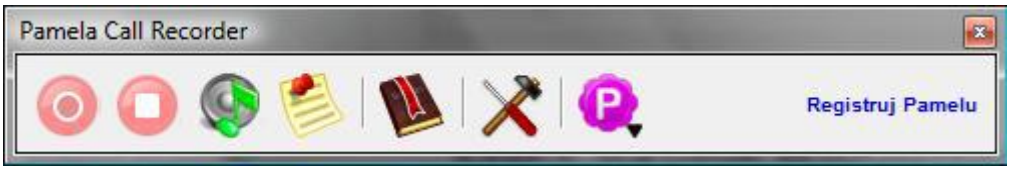
1.1.4 Dostupné programy pro nahrávání hovorů

Na internetu lze nalézt velké množství aplikací umožňující nahrávání hovoru programu Skype. V této části se budu snažit představit tři nejznámější programy. Každý z nich krátce

popíšu a uvedu jeho možnosti.

Pamela Skype recorder – Program se řadí mezi nejpokročilejší v tomto odvětví. Poskytuje velké množství funkcí pro záznam nejen běžného hovoru, ale i video hovoru. Záznamy lze ukládat do různých formátů. Taktéž podporuje ukládání historie textových zpráv. Uživatelské rozhraní aplikace je podobné jako u programu Skype. Výhodou je lokalizace programu do mnoha jazyků.

- Záznamenávání audio i video hovorů, konferencí
- Zaznamenávání historie chatu
- Automatické odpovědi
- Možnost přehrávat zvuky do hovoru
- Více formátu pro uložení zvuku (wave, mp3, wma, ogg)



Obrázek 1.4: Aplikace Pamela Skype recorder

MP3 Skype Recorder – Jak už z názvu programu vyplývá, program nahrává pouze běžný (hlasový) hovor do úsporného formátu MP3. Aplikace se zaměřuje na jednoduchost použití. Všechny ovládací prvky soustředí pouze do jednoho okna.

Obrázek 1.5: Aplikace MP3 Skype Recorder

- Zaznamenávání audio hovorů
- Formát záznamu - MP3
- Jednoduché a intuitivní uživatelské rozhraní

iFree Skype Recorder – Stejně jako předchozí program i tento je jednoduchý se změřením na snadné použití. Taktéž ukládá záznamy do souboru ve formátu MP3. Navíc umožňuje zasílat automaticky odpovědi na textové zprávy při nepřítomnosti uživatele.

- Zaznamenávání audio hovorů
- Formát záznamu - MP3
- Automatické odpovědi na chatu
- Jednoduché uživatelské rozhraní

Obrázek 1.6: Aplikace iFree Skype Recorder

1.2 Automatické rozpoznávání řeči

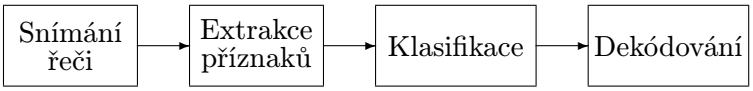
Automatické rozpoznávání řeči (angl. *Automatic speech recognition*) je technologie umožňující pomocí počítače převádět zvukovou promluvu do textové formy. Tento problém začali vědci zkoumat již kolem roku 1940. Od té doby postupně docházelo k výraznému pokroku v informačních technologiích. Tato, ale i jiné skutečnosti umožnily rozvinutí výzkumu řečových technologií. V dnešní době se stalo toto odvětví velmi perspektivní. Mnoho nových produktů společností z různých částí průmyslu využívá řečové technologie. Příkladem mohou být telefonní společnosti, webové společnosti, banky, soudy, policie a další.

Ideální rozpoznávač řeči by byl takový systém, který by byl schopný přesně rozpoznat všechna slova promluvy v co nejkratším čase. Přitom by nezáleželo na osobě, která promluvu pronesla ani na prostředí, ve kterém se promluva uskutečnila. Avšak realita ukazuje, že úspěšnost rozpoznávání závisí na velkém množství různých faktorů. Mezi ty, které nejvíce znesnadňují proces rozpoznávání, patří například - velmi velké slovníky pro některé jazyky, problémy plynulé řeči (poslední hláska slova je ovlivněná následujícím slovem, polykání koncových hlásek, apod.), hlučné prostředí promluvy a mnoho dalších [5].

V této části kapitoly se budu věnovat základním principům rozpoznávání řeči. Na začátku si představíme základní části, na které se rozpoznávání dělí. Následně je postupně rozebereme.

1.2.1 Principy rozpoznávání řeči

Rozpoznávání řeči sestává z několika oddělených procesů. Schéma základního principu rozpoznávání řeči je znázorněno na obr. 1.7. První část rozpoznávání zajišťuje snímání řečového signálu. Následuje rozdělení signálu na soubor malých částí (segmentů). Data segmentů jsou poté redukovaná na vektory příznaků, a ty potom klasifikovány. Výsledkem jsou slovní spojení. V následující odstavcích rozebereme jednotlivé části podrobněji.



Obrázek 1.7: Schéma rozpoznávání řeči

1.2.2 Snímání řečového signálu

Aby bylo možné analogový zvukový signál zpracovat počítačem, je nutné jej převést do digitální formy. Tuto konverzi zajišťuje A/D (Analogově digitální) převodník. Jeho funkce se dělí na dvě části:

- Vzorkování - je proces, v němž se snažíme získat konečný počet vzorků analogového signálu, který je dostatečný pro jeho reprodukci. Vzorky odebíráme s určitou frekvencí (vzorkovací frekvencí). O této frekvenci platí, že musí být dle *Shannonova* teorému nejméně dvojnásobně vyšší než nejvyšší frekvence vzorkovaného signálu. Pokud není toto pravidlo dodrženo, dojde k nenávratnému zkreslení signálu. Pro vzorkování lidského hlasu dostačuje frekvence 8 kHz.
- Kvantování - tento proces přiřazuje jednotlivým vzorkům hodnoty v konečném intervalu.

Pro účely rozpoznávání řeči je nutné signál rozdělit na krátké částí (segmenty), na nichž má signál stejné vlastnosti (je stacionární). Tento proces se nazývá segmentace. Délka segmentu se volí v rozsahu 10 až 30 ms. Pro přesnější zpracování se mohou rámce částečně překrývat. Na tyto rámce se dále aplikují některé procesy pro zlepšení jejích vlastností. Mezi tyto procesy patří například váhování segmentu Hammingovým oknem, ustředění signálu a jiné [4].

1.2.3 Extrakce příznaků

Digitální zvukový signál obsahuje velké množství dat. Všechna data jsou důležitá pro správnou reprodukci signálu. Avšak pro účely rozpoznávání řeči jsou důležité pouze některé vlastnosti. Kdyby měl klasifikátor rozpoznávat řeč ze všech dat signálu, byl by vysoce pomalý a neefektivní. Problémem je, že signál obsahuje mnoho redundantních dat, nedůležitých pro vlastní rozpoznávání. Cílem extrakce příznaků je redukovat velký počet dat na soubor vlastností (příznaku), jež tyto data popisuje co nejpřesněji z hlediska rozpoznávání řeči. Mezi nejběžnější metody, založenými na metodě krátkodobé analýzy jsou:

- LPCC (Linear Prediction Cepstral Coefficients)
- PLP (Perceptual Linear Prediction)
- MFCC (Mel Frequency Cepstral Coefficients)

V dnešní době se nejvíce využívá poslední zmiňovaná metoda MFCC, proto si ji blíže popíšeme. Proces získání MFCC příznaků pro jednotlivé rámce se dělí na tyto části:

- Váhování (nejčastěji Hammingovým oknem)
- Furierova transformace
- Aplikace melovských trojúhelníkových filtrů

- Logaritmus získaných hodnoty
- Diskrétní cosinova transformace

Pomocí zmíněného postupu získáme 12 kepstrálních koeficientů. Pro zvýšení kvality popisu řeči se k nim obvykle přidává další parametr – logaritmus krátkodobé energie celého rámce. K dalšímu rozšíření 13 parametrů lze použít první a druhou derivaci podle času jednotlivých koeficientů (tzv. Δ a $\Delta\Delta$ - koeficienty). Tím získáme dalších 26 parametrů. Cílová sada pak celkově obsahuje 39 koeficientů [4].

1.2.4 Klasifikace a dekódování

Po transformaci zvukového signálu na množinu vektorů příznaků následuje proces rozpoznání vlastního obsahu promluvy. Tento proces je výpočetně, časově i paměťově velmi náročný. Existuje několik postupů (algoritmů) pro řešení tohoto problému. V následujících odstavcích se seznámíme se dvěma základními přístupy řešení problému klasifikace.

Jedním se základních algoritmů pro řešení problému klasifikace jsou *skryté Makovovy modely (zk. HMM – Hidden Markov models)*. Tyto modely jsou založeny na statistice a vychází z teorie stavových automatů. Jednotlivé stavy modelů jsou prezentovány rozložením pravděpodobnosti částí signálu (např. fonémy). Markovovy modely jsou matematicky popsány pomocí přechodové matice A, kde jednotlivé prvky matice udávají pravděpodobnosti přechodů mezi jednotlivými stavy modelu, matice B – obsahující pravděpodobnosti, se kterými se bude proces vyskytovat v určitém stavu j, a rozložením pravděpodobnosti v počátečním stavu π [6].

Dalším klasickým řešením problému klasifikace je použití *neuronových sítí (Neural networks)*. Tato technologie do jisté míry modeluje nervový systém člověka. Síť sestává z velkého počtu uzlů (neuronů). Neurony jsou navzájem propojeny a organizovány do vrstev. Neuronová síť má vždy jednu vstupní a výstupní vrstvu. Mezi těmito vrstvami jsou další skryté vrstvy. Reakce každého uzlu je obvykle dána nelineární funkcí (např. logistická funkce) z váženého součtu jeho vstupů, přičemž optimální hodnoty vah uzlů sítě se zjistí při trénování modelu [5].

Kapitola 2

Knihovna pro komunikaci a nahrávání hovorů

Komunikaci externí aplikace s programem Skype zajišťuje knihovna, vytvořená speciálně pro tento případ. V této kapitole si ji rozebereme, přičemž se zaměříme na nejpodstatnější body návrhu, implementace a použití.

2.1 Návrh knihovny

Při návrhu knihovny jsem vycházel z postupů probraných v kapitole 1 v části 1.1. Knihovna zajišťuje komunikaci s programem Skype pomocí veřejného API. Dokumentace k tomuto API je dostupná z tohoto zdroje [1]. Při návrhu jsem se snažil o jednoduchost použití této knihovny. Problém jsem rozdělil do menších vzájemně oddělených částí. Každá s těchto částí řeší určitý druh problému. Části, na které jsem knihovnu rozdělil, lze vidět v následujícím seznamu:

- Interface pro komunikaci s programem Skype
- Zachycení a zaznamenání dat hovoru do zvoleného souboru
- TCP server pro příjem zvukových dat ze Skypu
- Modul pro chybové stavy a jejich hlášení

Jednotlivé moduly jsou mezi sebou propojeny a navzájem tvoří společný celek umožnující nahrávání hovorů. První část knihovny tvoří malý modul, který je schopen navázat a udržovat spojení s aplikací Skype, zasílat a přijímat zprávy a informovat uživatele o stavu komunikace.

Druhá část knihovny využívá interface pro komunikaci s programem Skype (první modul knihovny) a pomocí něj je schopná zjistit kdy začal a skončil hovor. V době, kdy hovor probíhá, může poslat prostřednictvím interface programu Skype příkaz, aby zvuková data přeposílal na zvolený port. Tam si je převezme třetí část knihovny, zpracuje je a uloží do zvoleného souboru.

Poslední čtvrtý modul knihovny obsahuje veškeré chybové stavy, které se mohou při jakékoli práci s knihovnou vyskytnout. Chyby se dělí na několik druhů (logovací zprávy, varování, běžné a závažné chyby). Modul má také funkci pro hlášení chyb. Tuto funkci si může uživatel nastavit.

Dále si rozebereme jednotlivé části knihovny a jejich funkce podrobněji.

2.1.1 Interface pro komunikaci s programem Sykpe

Teorií navázání a udržování spojení s aplikací Skype se zabývá kapitola 1. Knihovna zajišťuje komunikaci pouze na platformě Windows. Využívá pro to systém zasílání zpráv (Window messaging). Interface má tři základní funkce:

- Navázání a udržování spojení s programem Skype.
- Přijímání a zasílání zpráv.
- Informování o stavu komunikace.

Navázání spojení probíhá pomocí dvou zpráv. Jedna zpráva je žádost o navázání spojení, kterou posílá interface (klient) programu Skype. Druhou zprávu, která informuje o úspěchu navázání spojení, posílá program Skype zpět interface (klientovi). První zpráva obsahuje adresu nového vlákna (skrytého okna), které je vytvořeno před jejím odesláním. Druhá zpráva obsahuje adresu aplikace Skype. Po výměně svých adres může interface s programem Skype komunikovat. Skryté okno pak slouží pro zachycení zpráv zaslaných aplikací Skype.

Interface zajišťuje kontrolu stavu spojení. Kontrola je zajištěna prostřednictvím dvou příkazů veřejného API. Interface periodicky zasílá příkaz `PING` a očekává od aplikace Skype příkaz `PONG`. Jestliže odpověď nepřijde, spojení se automaticky ukončí. Interface posílá uživateli knihovny zprávy o stavu komunikace. Mohou nastat následující stavy:

- Spojení úspěšně navázáno.
- Interface čeká na potvrzení komunikace od uživatele.
- Komunikace explicitně zamítnuta uživatelem.
- Aplikace Skype není dostupná.
- Aplikace Skype je již dostupná.
- Spojení přerušeno.
- Příchozí zpráva (text zprávy).

2.1.2 Zachycení a zaznamenání dat hovoru

Tato část knihovny používá ke své činnosti interface popsaný v předchozí kapitole. Modul sleduje příchozí zprávy z aplikace Skype a hledá ty příkazy, které označují začátek a konec hovoru.

```
-> CALL 1425 STATUS INPROGRESS // začátek hovoru
-> CALL 1425 STATUS FINISHED // konec hovoru
```

Identifikuje-li modul začátek hovoru, je připraven na nahrávání. V případě, že uživatel spustí nahrávání, zašle modul pomocí interface aplikaci Skype zprávy pro nastavení přeposílání zvukových dat na zvolený port přes TCP/IP protokol. Data nahrávky se dělí na dva kanály. Prvním kanálem přichází zvuková data z mikrofonu, druhým data z internetu. Přeposílání dat je nutné nastavit pro oba kanály zvlášť. Slouží k tomu tyto dva příkazy:

```
<- ALTER CALL <id> SET_CAPTURE_MIC PORT="č_portu"
<- ALTER CALL <id> SET_OUTPUT PORT="č_portu"
```

Modul obsahuje dva TCP/IP servery popsané v následující kapitole. Po zaslání výše zmíněných příkazů začne aplikace Skype přeposílat data na zvolená čísla portů. Data přijímají TCP/IP servery. Data z obou serveru jsou sloučena a výsledek se ukládá do zvoleného souboru. Po ukončení nahrávání modul soubor uzavře. V průběhu hovoru může uživatel kdykoli spustit a vypnout nahrávání. Uživatel knihovny je prostřednictvím tohoto modulu informován o následujících stavech:

- začátek hovoru
- začátek nahrávání
- konec nahrávání (obsahuje typ a délku hovoru, datum a čas, jméno volajícího, název souboru s nahrávkou)
- konec hovoru

2.1.3 TCP server

Tento modul knihovny zajišťuje klasický TCP/IP server. Hlavní možnosti modulu jsou:

- Vytvoření TCP/IP serveru, který přijímá data na určité adrese a portu.
- Uživatelem definované zpracování příchozích dat.

TCP/IP server se v knihovně využívá k příjmu zvukových dat od programu Skype. Data přicházejí po paketech různé velikosti. Pakety obsahují surová zvuková data ve formátu RAW PCM se vzorkovací frekvencí 16 kHz a 16 bity pro jeden vzorek.

2.1.4 Modul pro chybové stavy a jejich hlášení

Modul zajišťuje hlášení všech chyb, které mohou v knihovně nastat. Chybové kódy jsou rozděleny podle jednotlivých modulů (např. chybové kódy modulu TCP začínají od čísla 300, chybové kódy interface modulu od čísla 100). Mohou nastat tři typy chyb:

- Logovací zprávy (např. TCP server připojen/odpojen).
- Varování (např. Na disku nemusí být dostatek místa pro uložení hovoru).
- Chyby (např. Nelze vytvořit soubor).

Modul dává uživateli možnost zvolit si zobrazování chybových hlášení. Je na něm, zda si hlášení nechá vypisovat do terminálu (např. v případě konzolové aplikace), nebo do grafického rozhraní v podobě informačních oken.

2.2 Implementace knihovny

Implementace knihovny je realizována pomocí jazyka C++. Při návrhu byl využit objektově orientovaný přístup. Jednotlivé moduly tak, jak byli popsány v předchozí kapitole, jsou rozděleny do jednotlivých tříd. Knihovna tedy obsahuje následující třídy.

- `SSkypeInterface` – Implementuje komunikaci mezi externí aplikaci a Skypem.
- `SSkypeRecorder` – Implementuje nahrávání hovorů .
- `STCP` – Implementuje TCP/IP server .
- `SSkypeErrHandlerI` – Čistě virtuální třída pro hlášení chybových stavů.

Hlavní třídou zajišťující nahrávání hovoru ze Skypu je `SSkypeRecorder`. Třída ke svojí činnosti využívá jedné instance `SSkypeInterface` (pro komunikaci s aplikaci Skype) a dvou instancí tříd `STCP` (pro získání zvukových dat nahrávky). Pro informování uživatele o stavech, ve kterých se třída `SSkypeRecorder` nachází se využívají tzv. zpětných volání (callback). Knihovna poskytuje speciální virtuální třídu `SSkypeRecoredCallback`, jejíž chování si nadefinuje uživatel a její instanci předá instanci třídy `SSkypeRecorder`. Virtuální metody třídy `SSkypeRecoredCallback` je možné vidět v následujícím seznamu:

- `OnInterfaceStatus(status)` – Informuje o stavu interfacu. Aktuální stav je předán v parametru metody.
- `OnStartCall()` – Signalizuje začátek hovoru
- `OnStopCall()` – Signalizuje konec hovoru.
- `OnStartCapture()` – Signalizuje začátek nahrávání.
- `OnStopCapture(typ, uživatel, soubor, začátek, délka)` – Signalizuje konec nahrávání. V parametrech metody jsou uloženy informace o nahrávce.

Třída `SSkypeInterface` využívá pro svou činnost Windows API. Toto API (Application Programming Interface) vyvinuté firmou Microsoft pro operační systém Microsoft Windows obsahuje základní funkce systému pro komunikaci mezi aplikacemi i funkce pro vytváření GUI (Graphical User Interface). Třída využívá zasílaní zpráv mezi okny (window messiging) pro komunikaci s aplikací Skype.

2.3 Kompilace a použití knihovny

Jak bylo zmíněno v minulé kapitole, knihovna využívá prostředky Windows API, tudíž lze knihovna přeložit pouze na této platformě. Pro překlad knihovny je nutné mít nainstalovaný překladač `MinGW`. Knihovna obsahuje soubor `Makefile` pro automatizaci překladu pomocí programu `make`. Výstupem překladu je statická knihovna `skyperec.a`. Použití knihovny lze shrnout do těchto bodů:

1. Připojení hlavičkového souboru `skyperec.h`.
2. Vytvoření instance třídy `SSkypeRecorder`.

3. Definování chování tříd `SSkypeRecoredCallback` a `SSkypeErrHandlerI`.
4. Vytvoření instancí definovaných tříd a předaní těchto instancí instanci třídy `SSkypeRecorder`.
5. Připojení k aplikaci Skype pomocí metody `Connect()`.
6. Nastavení cesty k souboru pro uložení nahrávky pomocí metody `SetSavePath(cesta)`.
7. Spuštění a ukončení nahrávání pomocí metod `StartCapture()` a `StopCapture()`.
8. Přilinkování statické knihovny `skyperec.a` k výsledné aplikaci.

Knihovna byla překládána překladačem `MinGW g++` ve verzi 4.4.5. Testování a ladění probíhalo na systému Microsoft Windows Vista s aplikací Skype (verze 5.1).

Kapitola 3

Aplikace Personal Voice Assistant

V předchozí kapitole jsme si popsali knihovnu zajišťující komunikaci s aplikací Skype a nahrávání hovorů. Aby uživatel mohl jednoduše pořizovat nahrávky svých hovorů, vytvořil jsem grafickou uživatelskou aplikaci s názvem „*Personal Voice Assistant*“. Program představuje jednoduchý nástroj pro pořizování, správu a filtrování nahrávek hovorů. Navíc obsahuje speciální možnost vyhledávání klíčových slov v nahrávkách.

V této kapitole bude rozebrán návrh, možnosti a implementace této aplikace. V návrhu se zaměříme zejména na koncepci uživatelského rozhraní, jaké cíle byly stanoveny a jakým způsobem se podařilo těchto cílů dosáhnout. Také si probereme použité nástroje pro tvorbu uživatelského prostředí. V dalších částech kapitoly bude řeč o hlavních možnostech aplikace. Pro lepší představu budou některé části obsahovat obrázky znázorňující použití jednotlivých možností. Dále bude u každé možnosti vždy uvedena její implementace a použité knihovny.

3.1 Návrh aplikace a grafického uživatelského rozhraní

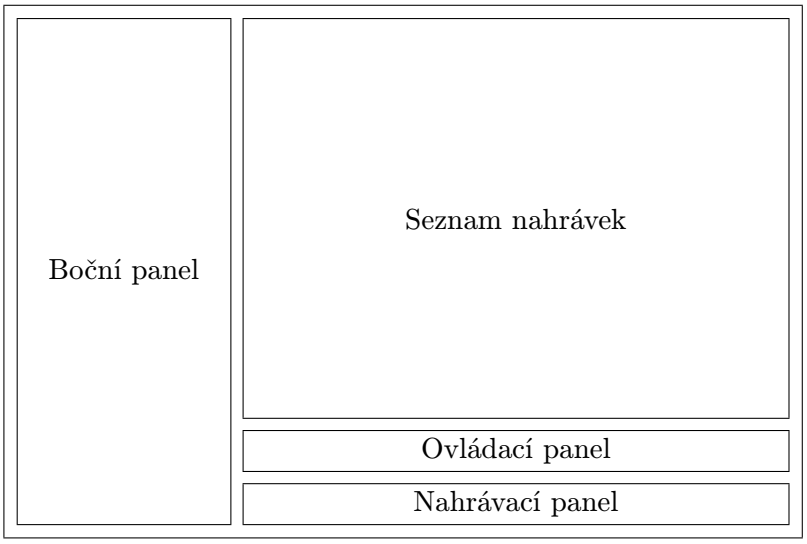
V této části kapitoly si rozdělíme aplikaci na jednotlivé části a stručně si je popíšeme. Zároveň si ukážeme postup návrhu grafického uživatelského rozhraní. Cílem návrhu aplikace bylo vytvořit co nejjednodušší nastroj pro zaznamenávání hovorů, tak aby byl srozumitelný a intuitivně se ovládal. Program obsahuje několik základních možností, podle kterých jsem návrh aplikace rozdělil. Části, na které jsem aplikaci rozdělil, je možné vidět v následujícím seznamu:

- Grafické uživatelské rozhraní
- Zobrazení nahrávek hovorů a filtrování.
- Přehrávání nahrávek hovorů.
- Zobrazení nahrávky ve Waveform editoru.
- Vyhledávání klíčových slov v nahrávce.

První čast, grafické uživatelské rozhraní, využívají ke svému ovládání ostatní části. Nyní si tuto část projdeme podrobněji.

Návrh grafického uživatelského rozhraní je důležitou součástí vývoje každé desktopové aplikace. Pří navrhování rozhraní jsem se zaměřil na jednoduchost a snadnost použití. Nechtěl jsem uživatele zbytečně zahltit mnoha vnořenými okny a zbytečným nastavením. Proto

jsem zvolil pouze jediné hlavní okno programu, které jsem rozdělil do několika části. Schéma rozdělení hlavního okna je vidět na obrázku 3.1. Hlavním prvkem okna je seznam nahrávek



Obrázek 3.1: Schéma rozdělení hlavního okna aplikace

hovorů. Ten obsahuje přehlednou tabulku záznamů s jejich atributy. Pomocí ovládacího panelu lze nahrávky mazat, upravovat, zobrazovat v externím editoru nebo přehrávat. Pod ovládacím panel se nalézá panel pro záznam hovoru. Tento panel informuje uživatele o tom, v jakém stavu je komunikace s aplikací Skype (připojen, odpojen, čeká na autorizaci atp.) a také informuje, zda-li právě probíhá nějaký hovor. Probíhá-li hovor, může uživatel pomocí ovládacích prvků na tomto panelu spustit zaznamenávání. Poslední částí okna je boční panel. Tento panel slouží pro různé účely. V záhlaví je opatřen ikonami pro přepínání jednotlivých funkcí. První funkcí je filtrování seznamu nahrávek. Druhou funkcí tohoto panelu je vyhledávání klíčových slov ve zvolené nahrávce.

Pro tvorbu grafického uživatelského rozhraní je využita multiplatformní C++ knihovna *wxWidgets*. Tato knihovna je aktivně vyvíjena od roku 1992. Za počátkem vývoje stál vývojář *Julian Smart* z univerzity v Edinburgu. Knihovna wxWidgets se řadí mezi nativní knihovny pro tvorbu grafického uživatelského rozhraní. To znamená, že se nesnaží napodobovat jednotlivé prvky rozhraní na různých platformách, ale prvky vytváří pomocí nativních funkcí. Důsledkem je nativní vzhled na všech platformách [8]. Tato knihovna byla zvolena s pozdějším záměrem rozšířit aplikaci na ostatní platformy.

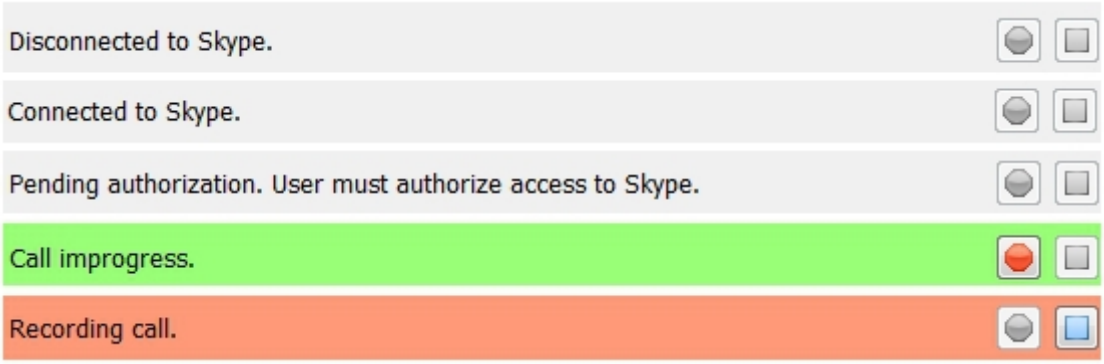
Pro návrh rozhraní pomocí knihovny *wxWidgets* je využit volně dostupný nástroj *wxForm builder*. Tento program se řadí mezi tzv. WYSIWYG aplikace. Jednoduše si navrhnete vzhled vaší aplikace a necháte si vygenerovat C++ kód, který požadovaný vzhled realizuje. Výsledný kód obsahuje třídy pro jednotlivá okna aplikace. Pro definování chování takto vytvořených tříd (oken) se využívá objektově orientovaného mechanismu dědičnosti, který je součástí jazyka C++. Třída definující chovaní okna aplikace jednoduše zdědí vlastnosti třídy definující vzhled.

Implementace rozložení grafických prvků aplikace je uchována v několika třídách vygenerovaných pomocí nástroje *wxForm builder*. Třída `MainWindowGui` obsahuje vzhled hlavního okna. Jeho chování pak ve třídě `MainWindow`. Další dvě třídy `FilterPanelGui` a `SearchPanelGui` realizují vzhled panelů pro filtrování nahrávek a pro vyhledávání klíčových

slov ve zvolené nahrávce. Jejich chování implementují třídy `FilterPanel` a `SearchPanel`. Poslední třídou je `RecordEditDialogGui`, která definuje vzhled okna dialogu pro úpravu názvu souboru nahrávky. Opět její chování definuje třída `RecordEditDialog`.

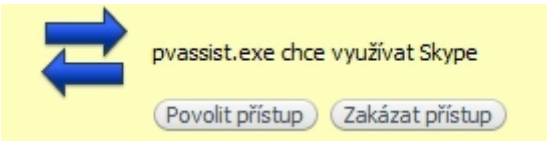
3.2 Nahrávání hovorů

První základní funkcí aplikace je nahrávání hovorů. K této činnosti využívá knihovnu pro komunikaci s aplikací Skype a nahrávání hovorů popsanou v kapitole 2. Knihovna poskytuje rozhraní, pomocí něhož se lze jednoduše připojit k programu Skype a nahrávat hovory. Informace o stavech, ve kterých se komunikace mezi aplikaci a programem Skype nachází, je zobrazována na speciálním panelu v dolní části aplikace. Panel obsahuje textové pole, do kterého jsou vypisována hlášení o aktuálním stavu. Dale panel obsahuje dvě tlačítka. První slouží ke spuštění nahrávání hovoru, druhé pro ukončení. Pro lepší upozornění má panel v době, kdy probíhá hovor zelené pozadí. Tím dává najevo svou připravenost k zaznamenávání hovoru. Při spuštění nahráváni se pozadí panelu zbarví do červena a tím signalizuje, že probíhá nahrávání. Jednotlivé stavy jsou znázorněny na obrázku 3.2 Přistup externí



Obrázek 3.2: Jednotlivé stavy panelu pro nahrávání hovorů

aplikace k programu Skype je nutné explicitně povolit. Pří spuštění externí aplikace se vás Skype dotáže, zda-li chcete aplikaci přistup povolit či zamítnout. Dotaz je vidět na obrázku 3.3. Zároveň se na panelu pro nahrávání zobrazí toto hlášení „Pending authorization. User must authorize access to Skype“. V případě, že uživatel přístup zakáže, na panelu se zobrazí hláška „The access to Skype was refused by user“. V takovémto případě není možno nahrávat žádný hovor, protože komunikace aplikace s programem Skype je zakázaná.



Obrázek 3.3: Dotaz programu Skype na potvrzení autorizace externí aplikace

Implementace nahrávání je tvořená jedinou třídou `SkypeCtrl`, která využívá instanci třídy `SSkypeRecorder`. Třída `SkypeCtrl` dědí z čistě virtuální třídy `SSkypeRecorderCallbackI` a implementuje jednotlivé stavy komunikace (začátek hovoru, konec hovoru, a jiné).

3.3 Zobrazení nahrávek hovorů a filtrování

Druhou základní funkcí aplikace je zobrazování pořízených nahrávek v přehledném seznamu. Tento seznam zabírá největší plochu aplikace. Díky tomu jsou přehledně vidět veškeré atributy nahrávky. Každý záznam má těchto 5 atributů:

- Typ hovoru (příchozí/odchozí).
- Jméno uživatele se kterým byl hovor uskutečněn (např. Jarda).
- Název souboru v němž je nahrávka uložena (např. `jarda_2001_3_26_15_15.wav`).
- Datum a čas kdy byl hovor uskutečněn.
- Trvání hovoru.

Pomocí kontextového menu lze jednoduše záznamy mazat, upravovat nebo zobrazovat v editoru. Jednotlivé záznamy lze vybrat myší a přehrát pomocí ovládacího panelu. Další možností je vybraný záznam připravit pro vyhledávání klíčových slov v řeči. Seznam nahrávek v aplikaci ilustruje obrázek 3.4.

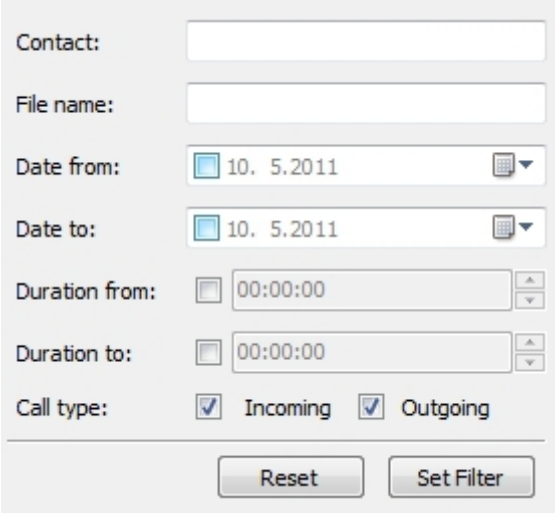
T...	Contact	File	Date	Duration
↑	Skype	echo123_2011_5_10_12_43_4.wav	05.10.2011 12:43:04	00:00:12
↑	Joseph	joseph74408_2011_3_26_15_5_15.wav	03.26.2011 15:05:15	00:00:50
↓	Šárka	sarinecka.v_2011_3_18_20_11_51.wav	03.18.2011 20:11:51	00:01:06
↑	Skype	echo123_2011_3_17_10_51_18.wav	03.17.2011 10:51:18	00:00:55
↑	Skype	echo123_2011_3_17_9_53_14.wav	03.17.2011 09:53:14	00:00:48
↑	Skype	echo_prejmenovany.wav	03.10.2011 18:54:40	00:00:13

Obrázek 3.4: Seznam zaznamenaný hovorů

Po určité době má uživatel velmi mnoho nahrávek. Aby mohl jednoduše najít to, co hledá, poskytuje aplikace dvě funkce, které tento proces usnadní. První je seřazení záznamů hovorů dle jejich libovolného parametru vzestupně i sestupně. Druhou je filtrování seznamu podle libovolných parametrů.

Možnost seřazení seznamu vidíte na obrázku 3.4. Zde jsou záznamy seřazeny vzestupně podle data uskutečnění nahrávky. Seřazení indikuje malá šipka v záhlaví seznamu u atributu, dle kterého se má řadit. Směr šipky určuje směr řazení. Výběr řazení dle jiného atributu lze realizovat kliknutím pravým tlačítkem na záhlaví sloupce s tímto parametrem. Při kliknutí na záhlaví sloupce již vybraného pro seřazení se změní směr řazení.

Filtrování seznamu nahrávek se nastavuje pomocí ovládacích prvků na bočním panelu na záložce s názvem „Filter“. Tato záložka obsahuje vstupní prvky, do kterých vložíme parametry, dle který chceme filtrovat a dvě tlačítka. Jedno pro nastavení filtru a druhé pro resetování filtru. Na obrázku 3.5 je možné vidět rozložení ovládacích prvků pro nastavení filtrace. Pro filtraci lze nastavit jméno kontaktu, se kterým byl hovor uskutečněn nebo jméno souboru. Přičemž lze použít znak * jako libovolný řetězec. Tudíž je možné napsat do filtru kontaktů například řetězec „m*“ a v seznamu budou zobrazeny pouze nahrávky s parametrem kontaktu začínající na písmeno m. Dále je možné záznamy filtrovat podle jejich data pořízení. Stačí nastavit počáteční a koncové datum a zobrazí se pouze nahrávky pořízené v tomto období. Taktéž lze nastavit časový interval pro filtraci záznamů s určitou délkou.



Obrázek 3.5: Rozložení ovládacích prvků pro nastavení filtrace

Poslední možností je filtrace podle typu. Lze tedy zobrazit pouze nahrávky příchozích nebo odchozích hovorů.

Implementaci jakéhokoliv seznamu poskytuje bázová třída `ListCtrlBase`. Tato třída je odvozená od třídy `wxListCtrl`, která je součástí knihovny wxWidgets. Bázová třída rozšiřuje možností základní knihovní třídy `wxListCtrl` o automatickou úpravu délek jednotlivých sloupců pří změně velikosti okna. Přičemž je možné nastavit sloupce s fixní délkou a sloupce roztažitelnou délkou. Dále umožňuje pomocí kontextového menu skrývání sloupců. Seřazení položek seznamu podle určitého atributu. Uložení a načtení seznamu ze souboru ve formátu xml a kódování UTF-8. K parsování xml souboru je použita volně dostupná knihovna *Tiny xml*. Tato knihovna poskytuje jednoduchý nástroj pro práci s xml soubory. Implementaci seznamu zaznamenaných hovorů zajišťuje třída `ListCtrlRecords`, která je odvozená od bázové třídy pro tvorbu seznamu. Tato třída nastavuje formát seznamu a jeho atributy. Dále přidává do kontextového menu možnost zobrazení nahrávky v editoru a přípravu pro vyhledávání klíčových slov. Taktéž přidává možnost filtrace. Této funkce využívá záložka pro filtrování umístěná na bočním panelu. Její implementace je ve třídě `FilterPanel`, která je odvozená od třídy `FilterPanelGui` generované programem pro tvorbu uživatelského rozhraní.

3.4 Přehrávání záznamů hovorů

Nezbytnou funkcí programu je přehrávání pořízených nahrávek hovorů. Tuto funkci zajišťuje panel pro přehrávání umístěný v dolní části aplikace. Kromě ovládacích prvků pro přehrávání záznamů jsou na panelu navíc dvě tlačítka pro ovládání seznamu nahrávek rozebraného v minule části kapitoly. Jedná se o tlačítka pro vymazání záznamu a editaci názvu souboru, ve které je nahrávka uložena. Následující obrázek 3.6 ukazuje podobu přehrávacího panelu.

Obrázek 3.6: Ukázka panelu pro přehrávání

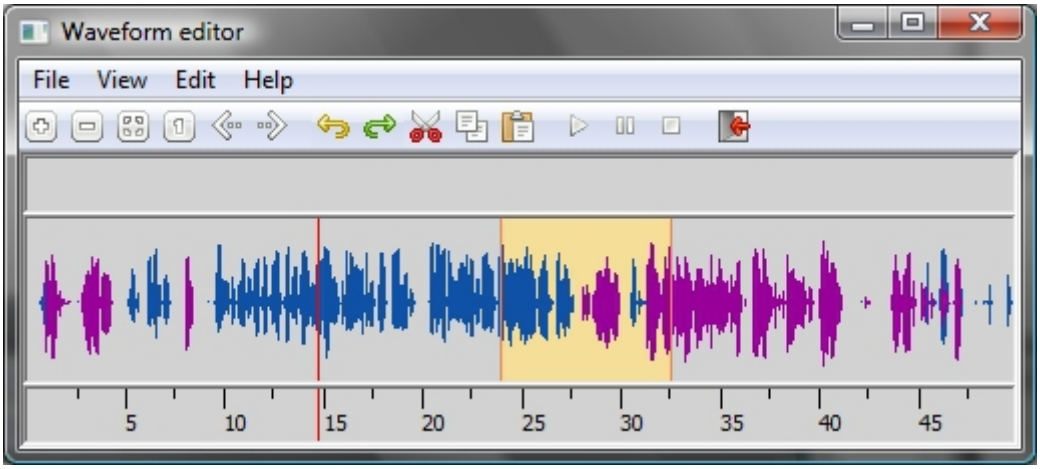
Z obrázku je vidět, že panel obsahuje standardní ovládací prvky pro přehrávání. Jedno tlačítko pro spuštění a pozastavení přehrávání. Druhé pro zastavení přehrávání. Dále je zde prvek pro zobrazení aktuální pozice, kterým se lze přeskakovat v čase záznamu. Poslední textový prvek doplňuje ukazatel pozice o jeho aktuální hodnotu.

Implementace přehrávání zaznamenaných hovorů je rozdělena do několika tříd. Základní tří třídy umožňující přehrání libovolného souboru ve formátu WAVE jsou ve zdrojových souborech `audio.h` a `audio.cpp`. První třída `WaveFile` umožňuje ze zvoleného zvukového souboru ve formátu WAVE přečíst hlavičku a na žádost načítat zvuková data o zvolené časové délce. Druhá třída `AudioRingBuf` zahrnuje implementaci kruhového zásobníku. Do zásobníku je možné ukládat data v jednom vlákně a v druhém data číst. Kruhový zásobník se používá jako vyrovnávací paměť pro zasílání dat na zvukovou kartu. Poslední třída `AudioOutput` využívá ke své činnosti instance obou předchozích tříd. Pro odesílání dat na zvukovou kartu využívá volně šířenou multiplatformní knihovnu *Port Audio*. Chování panelů na přehrávání zaznamenaných hovorů implementuje třída `SoundWidget`.

3.5 Zobrazení nahrávky ve Waveform editoru

V určitých chvílích potřebuje uživatel zaznamenaný hovor editovat. K tomu slouží funkce zobrazení aktuálně vybrané nahrávky v mnou navrženém externím zvukovém editoru *wEdit*. Umožňuje zobrazovat stopu nahrávky, přičemž dokáže barevně oddělit jednotlivé zvukové kanály. Zobrazovat pouze jednotlivé kanály. Dále umožňuje přibližování a oddalování grafu zvukové stopy (přiblížení až na jednotlivé vzorky), editaci zvukové stopy (mazání, kopírování vkládání), přehrávání vybrané části, zobrazení souboru s textovým přepisem (přípona .lab).

Editor *wEdit* je vyvíjen multi-platformě. Pracuje jak na operačním systému Microsoft Windows, tak na systémech odvozených ze systému Unix. Zobrazení aktuálně vybrané nahrávky hovoru v editoru je realizováno pomocí argumentu příkazové řádky Editoru. Zobrazení požadovaného souboru lze docílit zadáním parametru `-w` nebo `-input-wav` za nímž následuje cesta k souboru. Obrázek 3.7 ilustruje hlavní okno editoru.



Obrázek 3.7: Editor zvukových souborů wEdit

Implementace editoru je založena na multi-platformní knihovně *BSAPI* pro rozpoznávání řeči. Knihovna obsahuje modul pro vytvoření grafického editoru implementovaný libovolnou knihovnou pro tvorbu grafického uživatelského rozhraní. Požadavkem je pouze

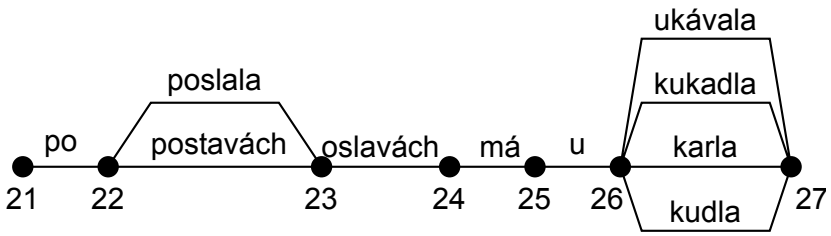
implementovat primitivní funkce pro kreslení (vykreslení bodu, čáry, boxu, textu). Aplikace *wEdit* využívá pro grafický výstup knihovnu *wxWidgets*.

3.6 Vyhledávání klíčových slov v nahrávce

Poslední významnou funkcí, která tuto aplikaci odlišuje od ostatních, je vyhledávání klíčových slov v záznamech hovorů. Funkce slouží ke snadnějšímu hledání v nahrávkách. Aby mohl uživatel vyhledávat klíčová slova, musí jako první připravit nahrávku pro vyhledávání. Poté jednoduše označí nahrávku, ve které se bude vyhledávat a zadá hledané klíčové slovo. Následné spustí vyhledávání. Aplikace vypíše do seznamu všechny výskyty hledaného slova, u nichž je zobrazen čas, ve kterém bylo slovo řečeno.

Vyhledávání klíčových slov je realizováno pomocí technologií vyvíjených *skupinou pro zpracování řeči* na fakultě informatiky VUT. Aplikace využívá systém LVCSR (zk. Large Vocabulary Conversational Speech Recognition), tedy systém pro rozpoznávání plynulé řeči založený na velkých slovnících. Pomocí této technologie je zvukový soubor s nahrávkou převeden do textové podoby. Výstupem mohou být textové soubory v různých formátech. Aplikace využívá pro uložení přepisu nahrávek a následné vyhledávání speciální formát *Confusion network*. Tento formát byl vyvinut pro ukládání přepisu řeči, přičemž přepis určitých časových úseků může obsahovat více alternativ. Formát je taktéž vhodný pro vyhledávání klíčových slov a informace v něm lze snadno indexovat. V následujícím příkladu je možné vidět několik řádků souboru v tomto formátu.

```
T=21 ST=11.10 ET=11.11 W=PO P=0.990206
T=22 ST=11.11 ET=11.50 W=POSLALA P=4.16415e-018
T=22 ST=11.11 ET=11.50 W=POSTAVÁCH P=4.71828e-025
T=23 ST=11.50 ET=11.94 W=OSLAVÁCH P=0.98618
T=24 ST=11.94 ET=12.16 W=MÁ P=0.986421
T=25 ST=12.16 ET=12.25 W=U P=0.000355055
T=26 ST=12.25 ET=12.81 W=UKÁZALA P=0.985458
T=26 ST=12.25 ET=12.81 W=KARLA P=0.000355185
T=26 ST=12.25 ET=12.81 W=UKÁVALA P=1.22949e-016
T=26 ST=12.25 ET=12.81 W=KUKADLA P=3.77165e-027
T=26 ST=12.25 ET=12.81 W=KADLA P=1.90912e-030
```



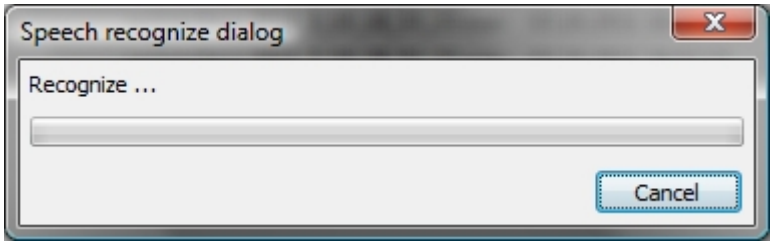
Obrázek 3.8: Confusion network

Na příkladu vidíte, že soubor obsahuje několik sloupců. První sloupec obsahuje číslo uzlu v grafu (znázorněno na obrázku). Ve druhé a třetím sloupci je počáteční a koncový čas slova. Předposlední obsahuje text samotného slova a poslední obsahuje číslo vyjadřující s jakou jistotou si je systém jistý, že mluvčí řekl právě toto slovo (čím vyšší číslo tím si je systém jistější). Na následujícím obrázku 3.8 je vidět graf přikadu řádku souboru.

Implementace vyhledávání klíčový slov lze rozdělit na dvě částí. Rozpoznání řeči pomocí systému LVCSR a vyhledávání klíčový slov ve výsledném souboru ve formátu popsaném výše. Pro rozpoznávání řeči jsem využil knihovnu *BSAPI* vyvíjenou skupinou rozpoznávaní řeči na FIT VUT a program `offlinerec.exe`, který je součástí této knihovny. Dokumentace ke knihovně je dostupná ze zdroje [2]. Program `offlinerec.exe` umožňuje rozpoznat řeč v audio souboru a výstup uložit do souboru s příponou cn (formát confusion network). Nevýhodou této technologie je její časová náročnost a vysoké nároky na výkon a paměť počítače. Program `offlinerec.exe` je konzolová aplikace, která se ovládá argumenty příkazové řádky. Skladba argumentů pro vystup s příponou cn je následující:

```
offlinerec.exe -c config_file -i in_file.wav -o out_file -f cn
```

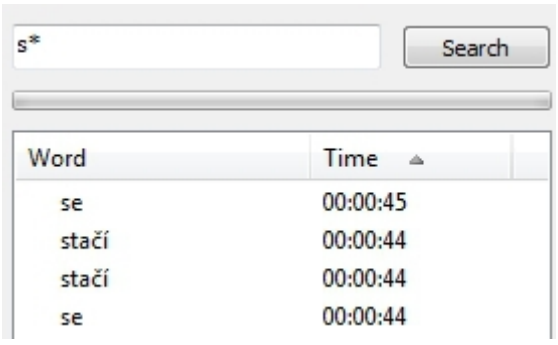
Parametr `-c` nastavuje cestu ke konfiguračnímu souboru. Následující dva argumenty `-i` a `-o` nastavují vstupní a výstupní soubory. Poslední parametr určuje formát výstupního textového souboru. Program s těmito parametry spouští aplikace Personal Voice Assistant pomocí metody knihovny *wxWidgets* `wxExecute()`. Při rozpoznávání řeči ve vybrané nahrávce se zobrazí dialogové okno informující uživatele o probíhajícím procesu. Dialogové okno je možné vidět na obrázku č. 3.9. Dialogové okno a volaní programu pro rozpoznávání



Obrázek 3.9: Dialogové okno zobrazené během rozpoznávání řeči v nahrávce

řeči je implementováno třídou `OfflinerecDialog`, která je odvozena ze třídy pro definici vzhledu `OfflinerecDialogGui`.

Po úspěšném dokončení rozpoznávání je možné v nahrávce vyhledávat klíčová slova. Ovládací prvky pro vyhledávání je možné nalézt na bočním panel na záložce pro vyhledávání. Záložka obsahuje jedno textové pole pro zadání hledaného klíčového slova, ukazatel



Obrázek 3.10: Panel pro vyhledávání klíčových slov

průběhu vyhledávání, seznam zobrazující výsledky vyhledávání a tlačítko pro odstartování vyhledávání. Podobu panelu pro vyhledávání je možné vidět na obrázku č. 3.10. Jak je vidět, je možné při vyhledávání použít znak hvězdičky pro libovolný řetězec.

Implementaci vyhledávání klíčových slov v souboru s příponou cn obsahuje třída z názvem `SearchPanel`, která má definovaný vzhled ve třídě `SearchPanelGui`. Vyhledávání pracuje ve vlákně, kde se jako první načtou slova ze souboru do paměti a následně se v nich vyhledává klíčové slovo.

Závěr

Cílem bakalářské práce bylo navrhnout aplikaci pro zaznamenávání hovorů programu Skype a pomocí řečových technologií umožnit uživateli v záznamech vyhledávat klíčová slova. Při návrhu a implementaci bylo nutné řešit různé problémy. Přesto mohu říci, že zadaný cíl se podařilo splnit. Důležité je však upozornit, že výsledek není v některých částech optimální a pro reálné využití je nutné pokračovat v dalším vývoji projektu.

První část projektu řešila problém komunikace externí aplikace s programem Skype a nahrávání hovorů. Problém se mi podařilo rozdělit na více menších části, které bylo následně jednodušší realizovat. Jako řešení problému vznikla knihovna v jazyce C++. Při návrhu jsem využíval objektově orientovaný přistup, který zajistil jednoduchost použití knihovny. Nevýhodou knihovny je možnost jejího použití pouze pro platformu Microsoft Windows. Proto by bylo vhodné v dalším vývoji rozšířit knihovnu na ostatní platformy. Přestože knihovna splnila vytyčené cíle, obsahuje pouze základní funkce. Další vývoj by je mohl rozšířit o další možnosti, mezi než patří zejména ukládání záznamů hovorů do různých formátů, schopnost zaznamenat audio konference nebo také zaznamenávat video hovory a historii textové komunikace.

Druhá část projektu se zabývala vytvořením grafické uživatelské aplikace využívající výše popsanou knihovnu. Při návrhu rozhraní aplikace jsem se snažil maximalizovat ergonomií ovládání. Tohoto cíle se mi podařilo dosáhnout jednoduchostí a přehledností návrhu hlavního okna, v čemž spočívá její hlavní výhoda. Aplikace byla vytvořena v jazyce C++ za pomocí multi-platformní knihovny *wxWidgets*. Poskytuje několik funkcí, mezi které patří zejména nahrávání, přehledné zobrazení, filtrování, řazení a přehrávání hovorů. Tyto základní možnosti ovšem poskytuje většina aplikací zabývající se podobným tématem. Mezi ty funkce, které dělají tuto aplikaci jedinečnou, patří zobrazení nahrávky v externím editoru zvukových souborů a vyhledávání klíčových slov v nahrávce. První zmíněná možnost je určena pro pokročilé uživatele, kteří mohou pomocí editoru prohlížet a upravovat zvukovou stopu. Druhá možnost, vyhledávání klíčových slov v nahrávce, usnadní rychlejší nalezení hledaných informací uživatelům s mnoha dlouhými záznamy hovorů. Bohužel implementace této funkce není zcela připravená pro reálné nasazení. Pro hledání klíčových slov v řeči se využívá systém LVCSR, který je velmi časově náročný a ke své činnosti potřebuje velké množství procesorového času a operační paměti. Čas potřebný pro přípravu funkce vyhledávání klíčových slov je srovnatelný s časem poslechu celého hovoru. A tak prozatím jedinou výhodou této funkce je skutečnost, že náročný proces rozpoznávání řeči je proveden pouze před prvním vyhledání, následné hledání klíčových slov trvá řadově sekundy. V dalším vývoji aplikace bych doporučil zaměřit se na vylepšení této funkce tak, aby byla pro uživatele co možná nejvíc přínosná. Zaměřil bych se hlavně na zpracování řeči během nahrávání nebo použití dalších metod rozpoznávání řeči jako je například systém KWS (Key Word Spotting).

Literatura

[1] Skype API reference. [online], 2009.
URL http://developer.skype.com/resources/public_api_ref.zip

[2] Brno Speech Core API Documentation. 2011.
URL http://www.phonexia.com/docs/bsapi/

[3] About Skype. [online], Naposledy navštíveno 2011.
URL http://about.skype.com/

[4] Chalupníček, K.: *Rozpoznávání - diktované řeči pro medicínské aplikace*. Diplomová práce, Vysoké učení technické, Fakulta informačních technologií, 2004.
URL http://www.fit.vutbr.cz/~chalupni/publikace/DP/DP.pdf

[5] Englund, C.: *Speech recognition in the JAS 39 Gripen aircraft - adaptation to speech at different G-loads*. Master thesis, School of Computer Science and Communication, Speech, Music and Hearing, 11th March 2004.
URL http://www.speech.kth.se/prod/publications/files/1664.pdf

[6] Rabiner, L.; Juang, B.: An introduction to hidden Markov models. *ASSP Magazine, IEEE*, ročník 3, č. 1, jan 1986: s. 4 – 16, ISSN 0740-7467, doi:10.1109/MASSP.1986.1165342.

[7] Telecompaper: Skype grows FY revenues 20%, reaches 663 mln users. [online], Naposledy navštíveno 8.3.2011.
URL http://www.telecompaper.com/news/skype-grows-fy-revenues-20-reaches-663-mln-users

[8] wxTeam: About the wxWidgets Project. 2011.
URL http://www.wxwidgets.org/about/